IN THE SPECIFICATION:

Please amend the following paragraphs as indicated:

[0013] Figure 1 is a block diagram illustrating one embodiment of the re-multiplexing module 100 according to the present invention. The remultiplexing module 100 can be used as one component in a packet processing system in conjunction with other functional modules, such as [[an]] a system controller or integrated backplane to create a reconfigurable processing element supporting high-speed transport of video, audio and data signals for digital cable television headends. In the description below, it is assumed that the signals are in an MPEG-2 format, but it is to be understood that the inventive system can be used to process signals in other compatible formats without departing from the scope of the invention. Further, although the example below receives six MPEG-2 input transport streams and generates two independent MPEG-2 output transport streams, the inventive system can support and generate any number of inputs and/or outputs.

[0036] Figure 4 is a block diagram illustrating the components of the output processor 124 in the inventive re-multiplexing module 100. In ~~general~~ the illustrated example, the output processor 124 is an FPGA (Field Programmable Gate Array) that conducts the required hardware tasks to generate two or more output streams from the data stored in the packet buffers 104. More particularly, the output processor 124 reads the selected packet data from the input packet buffers and/or the insert packet buffer 112, performs PID remapping, program clock reference (PCR) correction and any other desired or required packet editing, and may also insert new PID fields and other MPEG control information into the output streams as directed by the CPU. The output processing section then generates two or more

independent high-speed transport multiplex (HSTM) output streams incorporating the selected packet data. The output processing section also monitors the packet buffers 104, 112 to check for any newly arrived packets and to communicate to the host processor 114 of the presence of any new packets as well as any required packet identifying information. The output processing section also sends the packet data as a filtered packet stream to the message extraction portion 110. Each of the components in the output processor 124 will be described below with reference to Figure 4.

[0039] The bus control logic 400 generally controls the manner in which the packets are read from the input and insert packet buffers 104, 112. The reading process is generally conducted in three phase. The host processor's ISR signal is divided into 94 read slots, and each read slot is divided into three time domain multiplexed phases. Each phase contains one read cycle ~~form~~ from the packet buffer 104. The first phase includes the reading of data other than the packet data itself (e.g., timestamps, PID value, page number, PCR flag, CMP packet data). The second and third phases are used to read data stored in the packet buffers. These phases will be described in greater detail below.

[0040] The order in which the data is read during the first phase is dictated by the specific data needed to read the packet and the data's relative priority. In this example, timestamps have the highest priority and are read first. New packet header information is considered lower priority and CMP packet data is considered the lowest priority. The timestamps are assigned the highest priority in the first phase to ensure that the PCR correction calculation will be completed by the time the resultant data is to be inserted into the output packet data stream. As explained above, packets having a valid PCR filed are

detected and flagged by the input processor 120. Because the input processor 120 will report whether or not a given packet has a valid PCR, the bus control logic in the output processor only has to read the timestamps for packets identified as PCR packets, skipping over packets containing information other than a PCR. As a result, the first phase requires fewer timestamp reads than known processors, thereby increasing the speed of the CMP data output because it shares the same bus controller read phase. A prompt and quick CMP data output is desirable because delays in the CMP data output could cause a backup in the input packet buffer due to an unread packet, which can cause a buffer overflow error.

[0043] The second and third phases are used to read packet data stored in the input packet buffers and to write the data to packet data first-in-first-out (FIFO) buffers in the output processor 124. The second phase is used to generate a first output stream and the ~~second~~ third phase is used to generate a second output stream, assuming that the output processor generates two output streams. If the re-multiplexer is designed to generate more than two output streams, the bus control logic 400 causes the buffered data to be read in additional phases so that each phase corresponds to one output stream. Because the packet buffer read operations in this example read one word at a time, it will take about 94 reads to transfer an entire packet from the input packet buffer into the output transport stream. In this example, the data read out of the input packet buffers is based on the output stream data register 406 in the host processor interface 404.

[0047] As noted above, a packet data FIFO 410 is included in the front end of each output stream data path. Each FIFO [[140]] 410 is preferably relatively small (e.g., about 16 words deep) compared to the input packet buffers. The FIFOs 410 are included in the output

processor 124 because the time division multiplexing clock, which controls the phases of the read operations from the input packet buffers, and the output stream packet clock, which controls the data rate of the output multiplexing, are asynchronous. The data is read from the packet data FIFO 410 at a fraction of the output stream packet clock rate. The FIFO 410 may also include a "full" flag, such as a 3/4 full flag and/or a 1/2 full flag, which indicates to the bus control logic the fullness of the FIFO 410. The fullness of the FIFO 410 is monitored because the rate at which packets are sent to the output processor may be faster than the rate that the packets are output into the output stream. The 1/2 full flag may also be used by the output controller 414 to remove data to be inserted into the output data stream from the output processor 124.

[0050] First, the output controller 414 for each data stream sends an output to a packet data edit block 415. The packet data edit (PDE) circuitry 415 is used to overwrite data in the MPEG packet. More particularly, the PDE circuit edits the PID number, transport error indicator, and legal time window offset information. During this process, the output packet's PID value is replaced with a new PID value that is passed to each output processor output by the host processor. This PID replacement process is conducted for all of the packets sent to the output processor 124. Note that because the final PID remapping is performed in the output processor 124, after the packets have entered separate and independent data paths, one input packet ID stream can be assigned different PID values (i.e., they have unique identifiers) in each of the two output streams. Packets that are to be sent to the CMP 504 will not require PID re-mapping by the PDE circuitry because these packets will use the PID and source value, which identifies the source from which the packet is input, to identify the packets from the multiple input sources.

[0052] Software in the host processor 114 checks the fullness of the input packet buffers for any violations of re-multiplexer packet jitter specifications. If the input packet buffer 104 is filled to a predetermined level, the packets stored in the input buffers 104 are assumed to be corrupted. The re-multiplexing software may also check for other error conditions and can instruct the output processor to set the transport error indicator bit in the MPEG header in all packets associated with any one of the outputs ~~[why?]~~. Once the error condition has been resolved, the software will instruct the output processor 124 to resume normal packet processing ~~once the error condition has been resolved~~.

[0057] As mentioned above, the output processor 414 has a time reference generator 424. The output processor's time reference generator [[420]] 424 is the same as the time generator in the input processor except that the output processor time reference generator acts as the master counter that controls synchronization between the output processor 414 and the input processor. In one example, when the output processor's time reference generator 424 reaches a terminal count, it sends a synchronization load pulse to slave counters in the input processor 120 to reload all of the counters with 0's. The output timestamp is sampled when the first word is read from the FIFO.

[0065] The packet insertion function, which is performed by the output processor 124 and insert packet buffer 112 in conjunction with the host processor 114, inserts messages in any of the output streams. The message insertion function allows the re-multiplexer to produce MPEG compliant output transport streams while keeping its data processing load at a manageable level. Insert messages are sent to the re-multiplexer over the system bus. The

source of the messages will typically be the controlling element in the packet processing system (not shown). An insertion queue manager executing in software packetizes the messages and schedules the packet insertion into the output streams. In the re-multiplexer interrupt service routine, the software also checks for the availability of a packet whenever one of the output queues is empty. Insert packets are read the same way as the input packet data when [[the]] they are selected by the CPU. The insert packet buffer 112 itself can be a DPRAM that is divided into multiple blocks, one block associated with each output. In this example, the input packet buffer is an 8K x 16 DPRAM divided into two 4K blocks, one for each output.

[0077] A system interface 116 links the re-multiplexer module to other modules (not shown) in the MPS system via a parallel system bus. Information that can be sent to the re-multiplexer ~~include~~ includes initialization and configuration parameters, processing commands, and messages to be inserted into the output stream. Information that can be sent by the re-multiplexer to other MPS modules ~~include~~ includes the re-multiplexer module status, processing errors, and extracted messages. Note that, in practice, most of the communication will be between the re-multiplexer module and the system controller module.